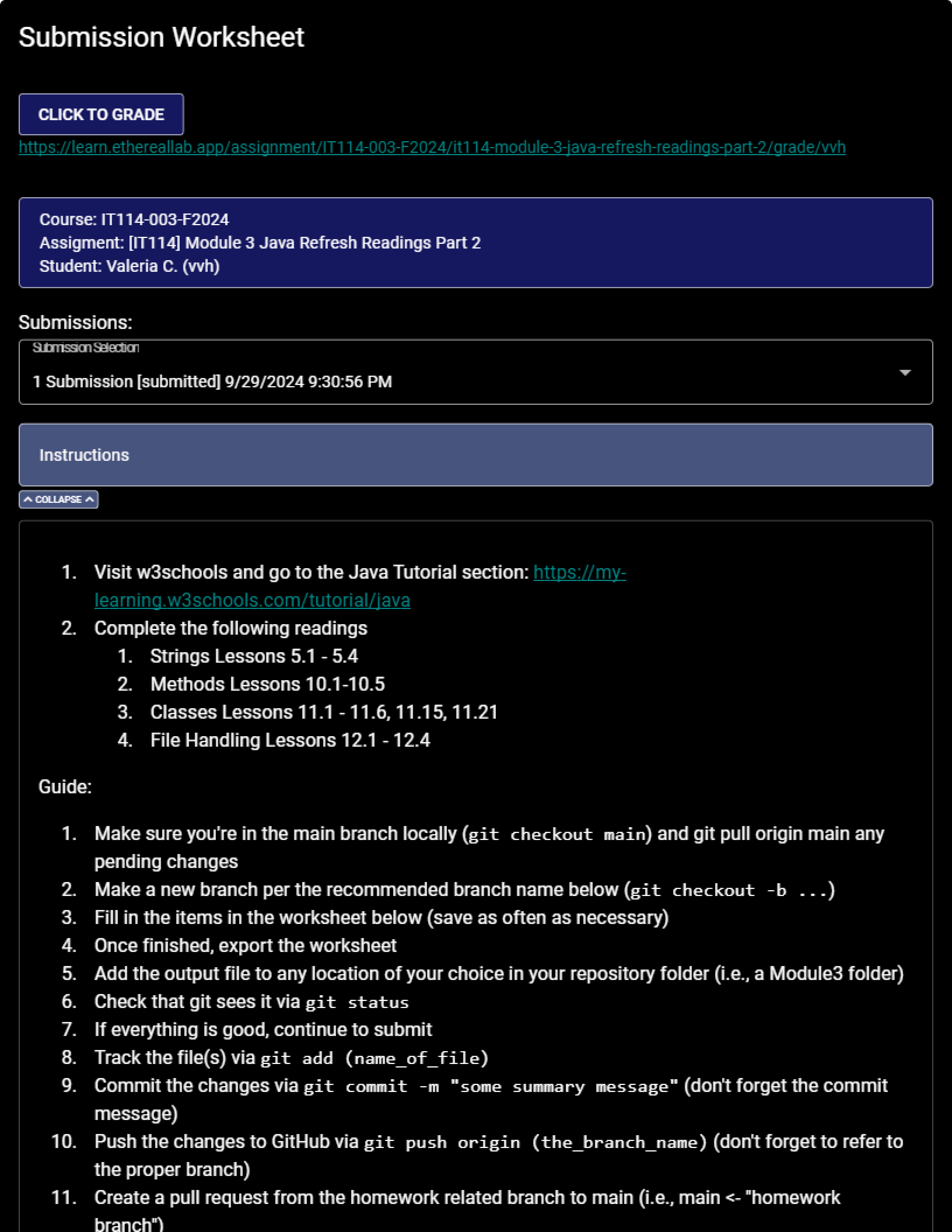


Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-module-3-java-refresh-readings-part-2/grade/vvh

Course: IT114-003-F2024
Assigment: [IT114] Module 3 Java Refresh Readings Part 2
Student: Valeria C. (vvh)

Submissions:

Submission Selection

1 Submission [submitted] 9/29/2024 9:30:56 PM

Instructions

COLLAPSE

1. Visit w3schools and go to the Java Tutorial section: https://my-learning.w3schools.com/tutorial/java
2. Complete the following readings
   1. Strings Lessons 5.1 - 5.4
   2. Methods Lessons 10.1-10.5
   3. Classes Lessons 11.1 - 11.6, 11.15, 11.21
   4. File Handling Lessons 12.1 - 12.4

Guide:

1. Make sure you're in the main branch locally (`git checkout main`) and git pull origin main any pending changes
2. Make a new branch per the recommended branch name below (`git checkout -b ...`)
3. Fill in the items in the worksheet below (save as often as necessary)
4. Once finished, export the worksheet
5. Add the output file to any location of your choice in your repository folder (i.e., a Module3 folder)
6. Check that git sees it via `git status`
7. If everything is good, continue to submit
8. Track the file(s) via `git add (name_of_file)`
9. Commit the changes via `git commit -m "some summary message"` (don't forget the commit message)
10. Push the changes to GitHub via `git push origin (the_branch_name)` (don't forget to refer to the proper branch)
11. Create a pull request from the homework related branch to main (i.e., main <- "homework branch")

12. Open and complete the merge of the pull request (it should turn purple)
13. Locally checkout main and pull the latest changes (to prepare for future work)
14. Take the same output file and upload it to Canvas

Branch name: M3-Java-Readings

Group

100%

Group: Learn Java Tutorial Part 2
Tasks: 1
Points: 8

COLLAPSE

Task

100%

Group: Learn Java Tutorial Part 2
Task #1: Read the following sections
Weight: ~100%
Points: ~8.00

COLLAPSE

Columns: 3

Sub-Task

100%

Group: Learn Java Tutorial Part 2
Task #1: Read the following sections
Sub Task #1: Strings Lessons 5.1 - 5.4

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Java Strings Screenshot

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Learn Java Tutorial Part 2
Task #1: Read the following sections
Sub Task #2: Methods Lessons 10.1-10.5

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Java Methods screenshot

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

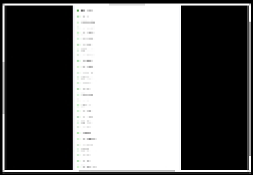
100%

Group: Learn Java Tutorial Part 2
Task #1: Read the following sections
Sub Task #3: Classes Lessons 11.1 - 11.6, 11.15,

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Java Classes Screenshot

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

Group: Learn Java
Tutorial Part 2

100%

Task #1: Read the following sections Sub Task #4: File Handling Lessons 12.1 - 12.4

Task Screenshots

Gallery Style: 2 Columns

4 2 1

File Handling Java Screenshot

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

End of Task 1

End of Group: Learn Java Tutorial Part 2
Task Status: 1/1

Group

100%

Group: Reflection
Tasks: 1
Points: 2

^ COLLAPSE ^

Task

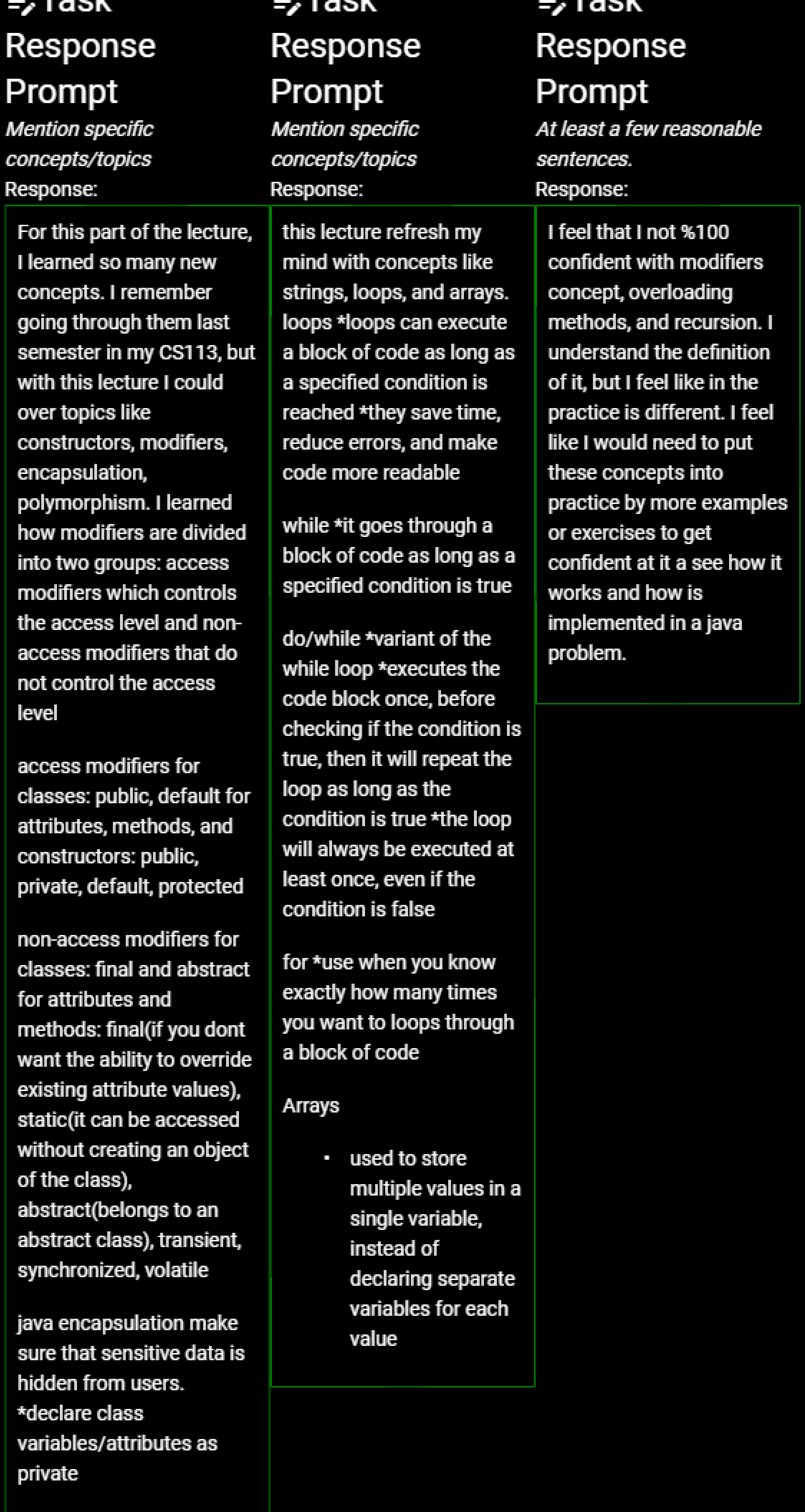
100%

Group: Reflection
Task #1: Reflect on the following topics
Weight: ~100%
Points: ~2.00

^ COLLAPSE ^

Columns: 4

Sub-Task 100% Group: Reflection Task #1: Reflect on the following topics	Sub-Task 100% Group: Reflection Task #1: Reflect on the following topics	Sub-Task 100% Group: Reflection Task #1: Reflect on the following topics
Task	Task	Task



Task

Response Prompt

Mention specific concepts/topics

Response:

For this part of the lecture, I learned so many new concepts. I remember going through them last semester in my CS113, but with this lecture I could over topics like constructors, modifiers, encapsulation, polymorphism. I learned how modifiers are divided into two groups: access modifiers which controls the access level and non-access modifiers that do not control the access level

access modifiers for classes: public, default for attributes, methods, and constructors: public, private, default, protected

non-access modifiers for classes: final and abstract for attributes and methods: final(if you dont want the ability to override existing attribute values), static(it can be accessed without creating an object of the class), abstract(belongs to an abstract class), transient, synchronized, volatile

java encapsulation make sure that sensitive data is hidden from users. *declare class variables/attributes as private

Task

Response Prompt

Mention specific concepts/topics

Response:

this lecture refresh my mind with concepts like strings, loops, and arrays. loops *loops can execute a block of code as long as a specified condition is reached *they save time, reduce errors, and make code more readable

while *it goes through a block of code as long as a specified condition is true

do/while *variant of the while loop *executes the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true *the loop will always be executed at least once, even if the condition is false

for *use when you know exactly how many times you want to loops through a block of code

Arrays

- used to store multiple values in a single variable, instead of declaring separate variables for each value

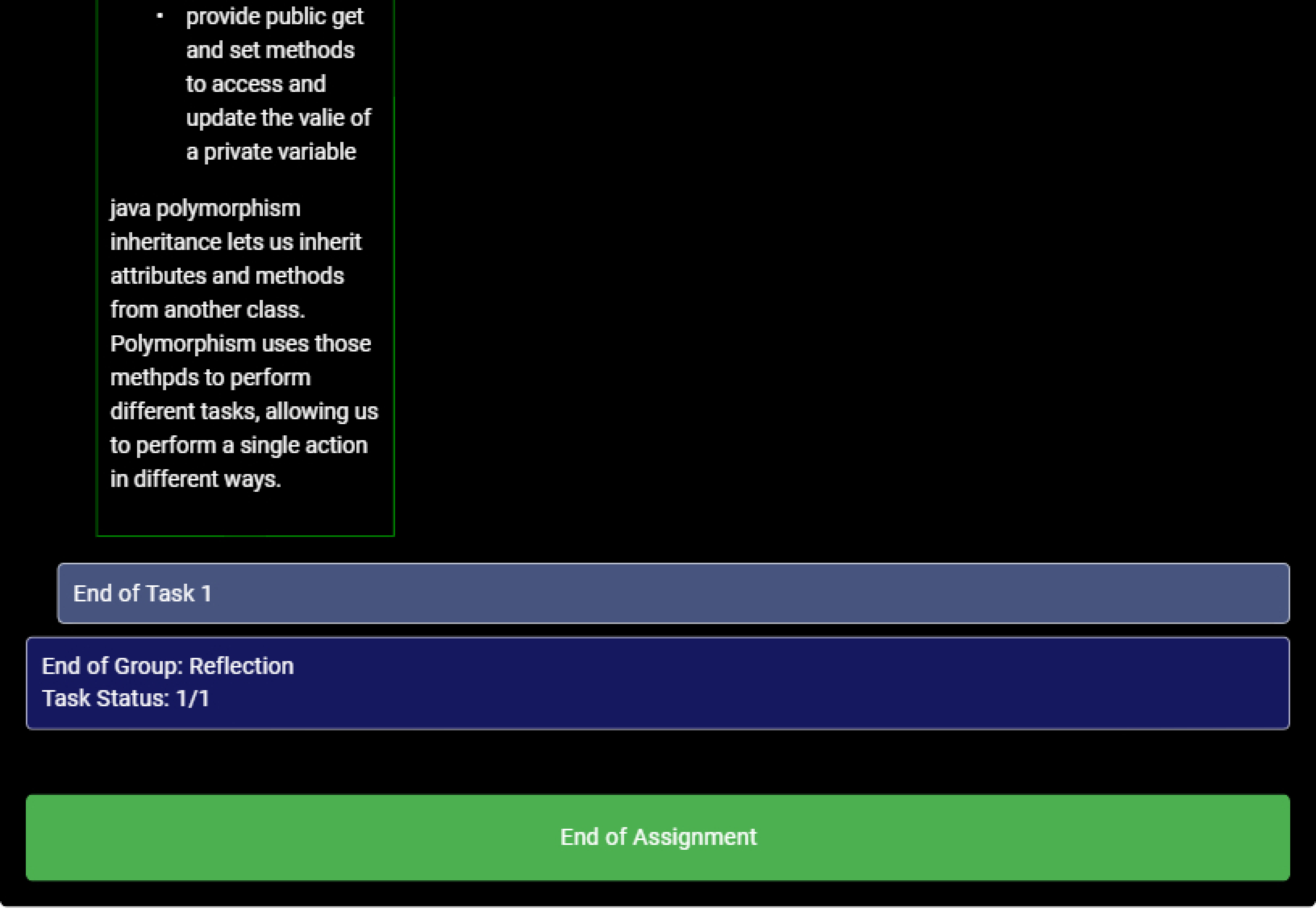
Task

Response Prompt

At least a few reasonable sentences.

Response:

I feel that I not %100 confident with modifiers concept, overloading methods, and recursion. I understand the definition of it, but I feel like in the practice is different. I feel like I would need to put these concepts into practice by more examples or exercises to get confident at it a see how it works and how is implemented in a java problem.



- provide public get and set methods to access and update the valie of a private variable

java polymorphism inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methpds to perform different tasks, allowing us to perform a single action in different ways.

End of Task 1

End of Group: Reflection
Task Status: 1/1

End of Assignment